Ejercicios de PowerShell: obtener información sobre el sistema operativo y enviarla por la red

Servidor

```
##Server
$port=2020
$endpoint = new-object System.Net.IPEndPoint ([IPAddress]::Any,$port)
$udpclient=new-Object System.Net.Sockets.UdpClient $port
$content=$udpclient.Receive([ref]$endpoint)
[Text.Encoding]::ASCII.GetString($content)
$udpclient.Dispose()
```

Cliente

```
Function menu {
    Clear-Host
    Write-Host "Iniciando el menu..."
    Write-Host "1. Información del hardware"
    Write-Host "2. Información de procesos"
    Write-Host "3. Ver registros de usuarios"
    Write-Host "4. Ver actualizaciones del sistema"
    Write-Host "5. Ver programas instalados"
    Write-Host "6. Salir"
}
1
menu

while($inp = Read-Host -Prompt "Escoge una accion"){

switch($inp){
         1{  $extension = Read-Host -Prompt "Pulse para analizar los componentes hardware de su pc"
```

```
            $port=2020
              $endpoint = new-object System.Net.IPEndPoint
([IPAddress]::Loopback,$port)
            $udpclient=new-Object System.Net.Sockets.UdpClient
$b=[Text.Encoding]::ASCII.GetBytes('$ComputerSystem=Get-
WmiObject Win32_ComputerSystem
            $BaseBoard=Get-WmiObject Win32_BaseBoard
            $BIOS=Get-WmiObject Win32_BIOS
            $Processor=Get-WmiObject Win32_Processor
            $Battery=Get-WmiObject Win32_Battery
            [PSCustomObject]@{
             Nombre=$Processor.Name
             Model = $ComputerSystem.Model
             Name = $ComputerSystem.Name
                                        PrimaryOwnerName   =
$ComputerSystem.primaryownername
                                    Totalphysicalmemory   =
$ComputerSystem.totalphysicalmemory
             ManufacturerBoard = $BaseBoard.Manufacturer
             BIOSVersion = $BIOS.SMbiosbiosversion
             BIOSSerialNumber = $BIOS.serialnumber
             version = $BaseBoard.version
             ManufacturerProcessor=$Processor.Manufacturer
             MaxClockSpeed=$Processor.MaxClockSpeed
             DeviceIDBattery=$Battery.DeviceID.trim()
             Designvoltaje=$Battery.designvoltage
             NumberOfCores=$Processor.NumberOfCores
NumberOfLogicalProcessors=$Processor.NumberOfLogicalProcessors
NumberOfEnabledCore=$Processor.NumberOfEnabledCore
             LoadPercentage=$Processor.LoadPercentage
             }')
$bytesSent=$udpclient.Send($b,$b.length,$endpoint)
             $udpclient.Close()
            pause;
            break
        }
        2
        {
              $extension = Read-Host -Prompt "Pulse para
analizar los procesos de su pc"
            $port=2020
```

```
             $endpoint = new-object System.Net.IPEndPoint
([IPAddress]::Loopback,$port)
            $udpclient=new-Object System.Net.Sockets.UdpClient
             $b=[Text.Encoding]::ASCII.GetBytes('Get-WmiObject
-Class win32_process | select name, Path, ExecutablePath,
CommandLine | Format-Custom')
            $bytesSent=$udpclient.Send($b,$b.length,$endpoint)
            $udpclient.Close()
            pause;
            break
        }
        3
        {
              $extension = Read-Host -Prompt "Introduzca el
nombre del equipo"
            $port=2020
              $endpoint = new-object System.Net.IPEndPoint
([IPAddress]::Loopback,$port)
            $udpclient=new-Object System.Net.Sockets.UdpClient
             $b=[Text.Encoding]::ASCII.GetBytes('$logs = get-
eventlog system -ComputerName "$extension" -source Microsoft-
Windows-Winlogon -After (Get-Date).AddDays(-7);
                  $res = @(); ForEach ($log in $logs)
{if($log.instanceid -eq 7001) {$type = "Logon"} Elseif
($log.instanceid -eq 7002){$type="Logoff"} Else {Continue}
$res += New-Object PSObject -Property @{Time =
$log.TimeWritten; "Event" = $type; User = (New-Object
System.Security.Principal.SecurityIdentifier
$Log.ReplacementStrings[1]).Translate([System.Security.Princip
al.NTAccount])}};
            $res')
            $bytesSent=$udpclient.Send($b,$b.length,$endpoint)
            $udpclient.Close()
        }

        6 {"Exit"; break}
           default {Write-Host -ForegroundColor red -
BackgroundColor white "Invalid option. Please select another
option";pause}
    }
```

```
menu
}
```